

PAT-NO: JP02000099397A  
DOCUMENT-IDENTIFIER: JP 2000099397 A  
TITLE: DATA PROCESSOR  
PUBN-DATE: April 7, 2000

**INVENTOR-INFORMATION:**

NAME	COUNTRY
ISHIKAWA, ATSUYOSHI	N/A
SAGESAKA, YASUHIRO	N/A
INOUE, MITSUYOSHI	N/A
KATAGIRI, MASASHI	N/A

**ASSIGNEE-INFORMATION:**

NAME	COUNTRY
HITACHI LTD	N/A
HITACHI ULSI SYSTEMS CO LTD	N/A
HITACHI YONEZAWA ELECTRONICS CO LTD	N/A

APPL-NO: JP10264103  
APPL-DATE: September 18, 1998

INT-CL (IPC): G06F012/06

**ABSTRACT:**

PROBLEM TO BE SOLVED: To realize the acceleration of a product-sum operation by parallelizing the readout of a multiplier and a multiplicand.

SOLUTION: This data processor adopts plural storage devices 103 and 104 which can be accessed by a central processor 101 and an offset adjusting device 102 which sets an offset address as rewritable and adds the offset address to an access address instead of the access address by a central processor in response to a read address by the central processor to output it, separately interfaces the storage devices to the central processor through a bi-directional data bus 112, introduces the access address from the central processor to a 1st storage device and gives an address from the offset adjusting device to a 2nd storage device. If the read address of the 1st storage device is operated by the offset adjusting device, data can parallelly be read from different addresses of both the storage devices.

Best Available Copy



## 【特許請求の範囲】

【請求項1】 中央処理装置と、前記中央処理装置によってリード・ライトアクセス可能な複数の記憶装置と、オフセットアドレスが書き換え可能に設定され前記中央処理装置による前記記憶装置に対するリードアクセスにตอบสนองして中央処理装置によるアクセスアドレスに代え当該アクセスアドレスに前記オフセットアドレスを加算したアドレスを出力するオフセット調整装置と、前記複数の記憶装置のデータ入出力端子を前記中央処理装置に個別に接続する双方向データバスと、前記中央処理装置によるアクセスアドレスを前記複数の記憶装置の内の一部の所定の記憶装置と前記オフセット調整装置に供給する第1のアドレスバスと、前記オフセット調整装置から出力されるアドレスを前記一部の所定の記憶装置以外の残りの記憶装置に供給する第2のアドレスバスと、を含んで成るものであることを特徴とするデータ処理装置。

【請求項2】 前記複数の記憶装置は2個であり、前記オフセット調整装置は、オフセットアドレスが書き換え可能に設定されるオフセットレジスタと、前記オフセットレジスタの設定値と前記中央処理装置からのアクセスアドレスとを加算する加算器と、前記加算器の出力又は前記第1のアドレスバスを選択して出力するセレクタと、前記記憶装置に対するリードアクセスにตอบสนองして前記セレクタに加算器の出力を選択させ、前記記憶装置に対するライトアクセスにตอบสนองして前記第1のアドレスバスを選択させるコントローラとを有して成るものであることを特徴とする請求項1記載のデータ処理装置。

【請求項3】 前記2個の記憶装置から並列的に読み出されたデータを並列に入力する積和演算装置を含んで成るものであることを特徴とする請求項2記載のデータ処理装置。

【請求項4】 1個の半導体チップに形成され、前記データバス及び前記第1のアドレスバスに接続され外部バスサイクルを起動可能なバスステートコントローラを更に含み、前記データバスは命令情報とデータ情報の双方の伝達に共用されるマイクロプロセッサとされて成るものであることを特徴とする請求項3記載のデータ処理装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、複数の記憶装置をリードアクセスしながらデータ処理を行うデータ処理装置に係り、例えば、デジタルフィルタや相関処理など積和演算が連続するデジタル信号処理の高速化処理に適用して有効な技術に関する。

## 【0002】

【従来の技術】 デジタル信号処理の特徴は、デジタルフィルタや高速フーリエ変換 (Fast Fourier Transform、以後FFTと言う)、相関処理にみられるように、

積和演算が数多く扱われることである。従来から、このような積和演算を高速に行うために、高速のハードウェア乗算器を内蔵し複数の内部バスを用いて高速パイプライン制御および多重オペレーションが可能なデジタル・シグナル・プロセッサ (Digital Signal Processor、以後DSPと言う) や専用LSIが用いられている。

【0003】 最近では、CPU (Central Processing Unit) に乗算機能やキャッシュメモリを内蔵した高速RISC (Reduced Instruction Set Computer) プロセッサを用いてデジタル信号処理を実現したものがある。例えば、そのようなデジタル信号処理として、静止画処理であるJPEG (Joint Photographic coding Experts Group) 処理や音声処理であるADPCM (Adaptive Differential PulseCode Modulation) 処理などが挙げられる。

【0004】 RISCプロセッサは一般的に命令の種類が少なく、かつ命令長が一定であるため、命令のデコード (解読) にハード・ワイヤード・ロジックを使うことが容易なため、高速なデコード処理が可能である。また、命令長を短くすることによって一度に複数の命令をフェッチ (読み込み) し後述するパイプライン処理の乱れを防止することができ、高速化が容易である利点がある。

【0005】 図6はパイプライン処理の一例を示したものである。図6において、中央処理装置が命令1から順次処理を進めていく場合、その処理はいくつかの段階を追って行われる。まず初めに命令の読み出しを行うフェッチ処理 (F) 501、次に命令を解読するデコード処理 (D) 502が行われる。最後に命令の実行処理

(E) 503が行われる。また、命令の種類によってはデータの読み出しや書き込みが必要になる場合、実行処理 (E) 503の直後にデータを参照するメモリアクセス処理 (M) 504が発生する。パイプライン処理では、例えば、4段階の処理で演算を行う場合、4つの命令がそれぞれ異なる段階で実行することによって効率良く処理することが可能である。図6の時刻T=5の時、命令5のフェッチ処理、命令4のデコード処理、命令3の実行処理が同時に実行されている。

【0006】 RISCプロセッサのバス構造はコスト低減のため命令バスとデータバスを共有したシングル・バス構造が一般的である。そのため、命令を読み出すフェッチ処理とデータを参照するメモリアクセス処理が同時 (同時刻) に発生するとバス競合が生じ、フェッチ処理またはメモリアクセス処理のいずれかが遅延され、パイプライン処理に乱れが生じる。図6の時刻T=7の時の命令4のメモリアクセス処理と時刻T=8の時の命令7のフェッチ処理が時刻T=7の時にバス競合を起こし、フェッチ処理が遅延した様子を示している。この競合を回避し、パイプライン処理の乱れを防止する手段として命令長の縮小化が考えられる。データバス幅を32ビット

トとし、命令長をバス幅の半分の16ビットに固定したとき、一度のフェッチ処理で前後2つの命令を一度に読み出せば、フェッチ処理が2回に1回の割合で発生するため、メモリアクセス処理とのバス競合の確率が低減する。図6は各命令の奇数番目に直後の命令も同時にフェッチする処理を行っており、偶数番目のフェッチ処理(f)はバスを占有しない。即ち、時刻T=6の時の命令6のフェッチ処理は既に完了しているため、命令3のメモリアクセス処理(M)とのバス競合は回避される。このように、バス競合を低減しパイプライン処理の乱れを防止することが、中央処理装置によるデータ処理の高速化および高効率化のために必要不可欠である。

【0007】図7は2つのデータのメモリアクセス処理が必要な場合のパイプライン処理の様子を示す。時刻T=5では、命令3に対するフェッチ処理601と、命令1に対するメモリアクセス処理が競合しないように、命令1の2番目のデータのメモリアクセス処理602が遅延する。同じ現象が時刻T=10、時刻T=15においても発生し、処理時間が長引く原因となっている。

【0008】ところで、前述のようにデジタル信号処理\*20

【数1】

$$\begin{aligned} \text{out}[i] &= \sum_{k=0}^n (\alpha[k] \cdot \text{in}[i-k]) \\ &= \alpha[n] \cdot \text{in}[i-n] + \alpha[n-1] \cdot \text{in}[i-(n-1)] + \\ &\quad \dots + \alpha[1] \cdot \text{in}[i-1] + \alpha[0] \cdot \text{in}[i] \end{aligned}$$

【0012】なお、上記演算は畳み込み(Convolution)演算と呼ばれ、係数 $\alpha[k]$ ( $0 \leq k \leq n$ ,  $n$ は0以上の整数)の与え方により何等かの周波数特性を持つデジタルフィルタを構成することができる。デジタル音声の高調波雑音を除去する低域通過(Low Pass)フィルタや音響用エコーキャンセラ、テレビの受信信号からゴースト信号を除去するゴーストキャンセラなど幅広く用いられている。

【0013】数1に示すように積和演算の回数はデジタルフィルタのタップ数 $n$ および入力されるデータ数に比例する。

【0014】また、一般的に、入力されるデータ $\text{in}$ は数十～数百のデータ数としてまとめて演算する場合が多く、入力データ列 $\text{in}$ や係数列 $\alpha$ 、出力されるデータ列 $\text{out}$ は、メモリ上でそれぞれ連続した配置構造を採るようにすることが多い。これにより、連続した各データブロックのアドレスポインタを設け、積和演算と同時に(同時刻)にアドレスポインタのインクリメント処理あるいはデクリメント処理を行えば、連続した積和演算を効率よく行える。即ち、アドレス演算によるオーバーヘッド処理がなくパイプライン処理の乱れが改善される。

【0015】

【発明が解決しようとする課題】前述のように、RISCプロセッサを用いてデジタル信号処理を行う場合、※50

\*理では積和演算が数多く扱われ、しかも連続した積和演算が多い。連続した積和演算をデジタルフィルタ処理を例に説明する。

【0009】図9は有限インパルス応答(Finite Impulse Response、FIR)フィルタ(以後、FIRフィルタと言う)のブロック図である。遅延素子801は所定のビット数のデータを記憶するレジスタであり、1サンプリング時間の固定遅延を行う。各遅延素子801から出力されるデータ(被乗数) $\text{in}[i] \sim \text{in}[i-n]$ は対応する各乗算器802に入力され、それぞれの乗算器802に設定される係数(乗数) $\alpha[0] \sim \alpha[n]$ と乗算され、加算器803に入力される。

【0010】図9のFIRフィルタで行われる演算は、サンプリング時間毎の時刻 $T=i$ の時の入力を $\text{in}[i]$ 、出力を $\text{out}[i]$ 、乗算器802で用いる係数をそれぞれ $\alpha[0] \sim \alpha[n]$ とすると、下記数1のように表すことができる。

【0011】

【数1】

※命令長を縮小しフェッチ処理回数を削減することによりバス競合を回避し、また、連続した積和演算を高速に実行するために乗算器を内蔵し、アドレス演算を積和演算と並列に実行することによりパイプライン処理の高速化が可能である。しかしながら、積和演算を実行するためには、少なくとも乗数、被乗数、命令の3種類のデータの読み出しが必要であり、これらを互いに競合することなくメモリから読み出す必要がある。前述の従来例では、フェッチ処理の削減には成功しているが乗数と被乗数の読み出しでは相変わらずパイプライン処理が乱れる場合があることが本発明者によって明らかにされた。

【0016】そこで、本発明者は、サンプリング時刻毎に乗数と被乗数をCPUによる同一アクセスサイクルでメモリから一緒に読み出すことについて検討した。また、乗数(係数)はメモリ上の一定のアドレスエリアに固定的に配置できるが、被乗数(入力データ)は、サンプリング時刻毎に順次蓄積され、最新のサンプリング時刻の被乗数データの格納アドレスは逐次変化されることになる。したがって、サンプリング時刻毎に被乗数を読み出すアドレスエリアはサンプリング時刻毎に変化される。一方、乗数のリードアドレスエリアは一定である。また、サンプリング時刻毎の被乗数データの書き込み動作は乗数データの配置とは無関係に行なわれなければならない。換言すれば、被乗数データの書き込み

際して乗数データと一緒に書き換えられることがあってはならない。逆に、インパルス応答特性を変化させるために乗数データを書き換えるとき一緒に被乗数データが書き換えられる事態は阻止しなければならない。

【0017】特開昭62-78641号公報には、キャッシュメモリ内の複数のデータアレイ内の複数のブロックのデータを加算器を経たアドレスと加算器を経ないアドレスとで一緒に読み出し可能にする技術が記載されている。加算器はアドレスにNを加算することができ、Nはデータアレイ内のブロックのバイト長であると記載されている。これによれば、リニアアドレスに連続して配置された複数ブロックに対して相隣接する複数ブロックと一緒にリードアクセスすることができる。

【0018】しかしながら、リードアクセスアドレスにデータブロックのバイト長を加算したアドレスを用いても、上記本発明者による検討事項を満足させることはできない。

【0019】本発明の目的は、格納アドレスエリアが逐次変化する複数の被乗数データと格納アドレスエリアが一定の複数の乗数データとを順番に同一アクセスサイクルでメモリから一緒に読み出すことができ、しかも、乗数データと被乗数データとを個別的に書替え可能なデータ処理装置を提供することにある。

【0020】本発明の別の目的は、積和演算処理のための乗数データと被乗数データとの読み出しによるパイプライン処理の乱れを低減できると共に、サンプリング時刻毎の被乗数データの書き込み動作を乗数データの配置アドレスとは無関係に行なうことができ、高速な積和演算を実現することができるデータ処理装置を提供することにある。

【0021】本発明の前記並びにその他の目的と新規な特徴は本明細書の記述及び添付図面から明らかになるであろう。

【0022】

【課題を解決するための手段】本願において開示される発明のうち代表的なものの概要を簡単に説明すれば下記の通りである。

【0023】すなわち、データ処理装置(100)は、中央処理装置(101)と、前記中央処理装置によってリード・ライトアクセス可能な複数個の記憶装置例えば第1及び第2の記憶装置(103、104)と、オフセットアドレスが書き換え可能に設定され前記中央処理装置によるリードアクセスにตอบสนองして中央処理装置によるアクセスアドレスに代え当該アクセスアドレスに前記オフセットアドレスを加算したアドレスを出力し、前記中央処理装置によるライトアクセスにตอบสนองして中央処理装置によるアクセスアドレスをそのまま出力するオフセット調整装置(102)と、前記複数個の記憶装置のデータ入出力端子を前記中央処理装置に個別に接続する双方向データバス(112)と、中央処理装置によるアクセ

スアドレスを前記複数個の記憶装置の内の一部の所定の記憶装置たとえば前記第1の記憶装置と前記オフセット調整装置とに供給する第1のアドレスバス(110)と、前記オフセット調整装置から出力されるアドレスを前記一部の所定の記憶装置以外の記憶装置たとえば前記第2の記憶装置に供給する第2のアドレスバス(111)とを含む。

【0024】上記により、第1の記憶装置のデータを参照するためのアドレスをオフセット調整装置により操作することにより、第1及び第2の記憶装置の異なるアドレスから中央処理装置にデータを並列に読み出すことが可能になる。例えば、図4に例示されるように、第1及び第2の記憶装置(103、104)の並列アクセスビット数が2バイトであって、アドレスの最下位がバイトアドレスを規定するものとし、メモリアドレスは中央処理装置のアドレス空間における絶対的なアドレスとして把握するものとする。時刻*i*における数1の積和演算を行う場合、アドレスオフセットをOFS1=10とすれば、アクセスアドレス\$100に対する中央処理装置のリードアクセスにより、アドレス\$100の乗数 $\alpha[0]$ とアドレス\$110の被乗数 $in[i]$ とを一緒にリードアクセスすることができ、以下アクセスアドレスを4ずつディクリメントしていけば、数1のout[i]を演算できる。

【0025】前記オフセットアドレスはプログラマブルに設定可能であるから、サンプリング時刻が異なる毎に、オフセットアドレスを変化させれば、時刻*i*の場合と同様に、順次乗数と被乗数を一緒にリードすることができる。例えば、図4の例において、時刻*i*+1における数1の積和演算を行う場合には、オフセットアドレスをOFS2=14に変更することにより、中央処理装置によるアクセスアドレス\$100に対するリードアクセスでアドレス\$100の乗数 $\alpha[0]$ とアドレス\$114の被乗数 $in[i+1]$ とを一緒にリードアクセスでき、中央処理装置によるアクセスアドレス\$96に対するリードアクセスでアドレス\$96の乗数 $\alpha[1]$ とアドレス\$110の被乗数 $in[i]$ とを一緒にリードアクセスすることができ、以下同様にアクセスアドレスを4ずつディクリメントしていけば、out[i+1]の結果を得ることができる。

【0026】オフセット調整装置による加算演算はリードアクセスにตอบสนองして選択され、ライトアクセスの場合には第1及び第2の双方の記憶装置には同じアドレス信号が供給され、アクセスアドレス等に従って第1及び第2の記憶装置は個別的にアクセス可能になる。例えば、図4の例において、時刻*i*+1の被乗数データ $in[i+1]$ を第2の記憶装置に書き込むとき、CPUは\$114をアクセスアドレスとしてライトアクセスを指示すれば、オフセット加算による並列アクセスは行なわれず、第2の記憶装置だけがライトアクセスされる。

【0027】上記により、積和演算のために必要な2つのデータ列のデータを並列に読み出すことができ、パイプライン処理の乱れを低減して積和演算を高速に処理することができる。

【0028】データ処理装置を1個の半導体チップに形成してRISCプロセッサとする場合、前記データバス及び前記第1のアドレスバスに接続され外部バスサイクルを起動可能なバーステートコントローラを含み、また、前記データバスを命令情報とデータ情報の伝送に共同させてバス構造の簡素化を図ることができる。

【0029】

【発明の実施の形態】図1には本発明に係るデータ処理装置の一例であるマイクロプロセッサが示される。同図に示されるマイクロプロセッサ100は、特に制限されないが、公知の半導体集積回路製造技術によって単結晶シリコンのような1個の半導体チップに形成されている。このマイクロプロセッサ100は、特に制限されないが、CPU100、オフセット調整装置102、第1の記憶装置103、第2の記憶装置104及びバーステートコントローラ105を有する。マイクロプロセッサ100は、内部バスとして第1のアドレスバス110、第2のアドレスバス111、双方向データバス（以下単にデータバスとも記す）112、コントロールバス113等を有する。データバス112は、特に制限されないが、32ビットのバス幅を有する。アドレスバス110、111に供給されるアドレスは最小単位がバイトアドレスを規定する。

【0030】前記CPU101は、特に制限されないが、算術論理演算器などを有する整数ユニット120と共に積和演算装置などを有するDSPユニット121を備える。DSPユニット121にはDSPレジスタ122が割り当てられ、整数ユニット120には汎用レジスタ123が割り当てられる。その他にCPU101には制御レジスタ124が設けられている。CPU101における命令フェッチ及び命令デコードなどの命令制御は命令制御部125が行なう。また、演算に必要なデータフェッチは命令制御部125から出力される命令デコード信号や制御信号に基づいてデータ制御部126が行なう。CPU101はバーステートコントローラ105を介して図示を省略する外部メモリなどから命令をフェッチし、その命令を命令制御部125の命令デコーダにて解説することにより、整数ユニット120やDSPユニット121を用いて当該命令に応じたデータ処理を行う。

【0031】前記バーステートコントローラ105は、CPU101によるアクセス対象回路（アクセス対象とされるアドレスエリア）に応じて、アクセスデータサイズ、アクセスタイム、ウェイトステートの挿入制御などを行なって、外部バスサイクルを制御する。

【0032】図2には前記記憶装置103、104の一

例が示される。前記記憶装置103、104は、夫々メモリ103M、104Mとデコーダ103D、104Dとを有し、DSPユニット121による積和演算のためのデータメモリとして利用される。記憶装置103、104のメモリ103M、104Mは夫々16ビットのデータ出力端子を有する。データバス112の下位16ビット（下位2バイト）D15～D0は第1の記憶装置103のメモリ103Mに、データバス112の上位16ビット（上位2バイト）D31～D16は第2の記憶装置104のメモリ104Mに接続される。この例では、メモリ103M、104Mは、同一の構成を有し、例えば、公知のSRAM（Static Random Access Memory：スタティック・ランダム・アクセス・メモリ）によって構成される。リード・ライト信号R/WはCPU101からコントロールバス113を介して第1及び第2の記憶装置103、104に共通に供給される。デコーダ103Dは、第1のアドレスバス110上のアドレスビットA1を入力し、A1=0のとき、メモリ103Mのメモリーネーブル信号ME1をイネーブル（活性）レベルにする。メモリ103Mはメモリーネーブル信号ME1がイネーブルレベルにされることによって動作可能になる。また、デコーダ104Dは、第2のアドレスバス110上のアドレスビットA1を入力し、A1=1のとき、メモリ104Mのメモリーネーブル信号ME2をイネーブル（活性）レベルにする。メモリ104Mはメモリーネーブル信号ME2がイネーブルレベルにされることによって動作可能になる。メモリ103Mはアドレス入力端子にバス110上のAn～A2を入力し、それをデコードしてワード単位のアドレッシング動作を行う。メモリ104Mはアドレス入力端子にバス111上のAn～A2を入力し、それをデコードしてワード単位のアドレッシング動作を行う。

【0033】前記オフセット調整回路102は、第1のアドレスバス110からアドレス信号を入力し、データバス112からオフセットアドレス（以下単にオフセット値とも記す）が供給され、コントロールバス113を介してリード・ライト信号R/Wが供給される。そして、オフセット調整回路102は第2のアドレスバス111にオフセットアドレスが加算されたアドレス信号を出力可能にする。第2のアドレスバス111に出力されたアドレス信号は第2の記憶装置104のアドレス入力端子に与えられる。

【0034】前記オフセット調整装置102は、オフセットアドレスが書き換え可能に設定されるオフセットレジスタ201と、前記オフセットレジスタ201の設定値と前記第1のアドレスバス110からのアクセスアドレスとを加算する加算器202と、前記加算器202の出力又は前記第1のアドレスバス110を選択して出力に伝達するセレクト203と、オフセットコントローラ204とを有する。オフセットコントローラ204は、

第1のアドレスバス110よりアドレス信号を入力すると共にコントロールバス113からリード・ライト信号R/Wを入力し、リードアクセスにตอบสนองして前記セクタ203に加算器202の出力を選択させ、ライトアクセスにตอบสนองして前記第1のアドレスバス110を選択させる選択制御論理を有する。更に、オフセットコントローラ204は、アドレスバス110からのアドレスをデコードし、オフセットレジスタ201に割当てられたアドレス信号を検出すると、そのときのライトアクセスに

10 ตอบสนองしてデータバス上のオフセットアドレスをオフセットレジスタ201に書き込み制御する。

【0035】次に、前記マイクロプロセッサ100を用いてFIRフィルタの畳み込み演算を行う場合について図4を参照しながら説明する。図4に示されるX、Y、ZなどはCPU101のアドレス空間における絶対的なアドレスを意味する。

【0036】前記数1において、時刻 $T=i$ の時の入力データを $in[i]$ 、出力データを $out[i]$ とする。また、フィルタ係数を $\alpha[k]$  ( $0 \leq k \leq n$ ,  $n$ は0以上の整数)とする。 $n$ はフィルタのタップ数である。

【0037】まず初めに、係数列 $\alpha$ を記憶装置103に配置し、入力データ列 $in$ を記憶装置104に配置する。演算結果 $out$ のデータ列は記憶装置104に記憶領域を確保する。また、各データ列の配置順序は $in$ および $out$ は昇順に配置し、 $\alpha$ は降順に配置する。ここで、時刻 $T=0$ の時の演算対象となる入力データ列は、数1によれば $in[-n]$ 、 $in[-n+1]$ 、 $\dots$ 、 $in[-1]$ 、 $in[0]$ となる。 $in[0]$ を除く入力データ列 $in[-n]$ 、 $in[-n+1]$ 、 $\dots$ 、 $in[-1]$ は、時刻 $T=0$ の時のフィルタ状態を意味し、それには何等かの

30 初期値が設定されることになる。

【0038】次に、係数列 $\alpha$ の先頭アドレスX、即ち $\alpha[n]$ の示すアドレスと入力データ列 $in$ の先頭アドレスY、即ち $in[-n]$ の示すアドレスの差分( $X-Y$ )をオフセット値の初期値とし、オフセット調整装置102のオフセットレジスタ206に保持する。出力データ列 $out$ の先頭アドレスをZとする。時刻 $T=0$ の時の出力データ $out[0]$ は数2の式で表される。

【0039】

40 【数2】  $out[0] = \alpha[n] \cdot in[-n] + \alpha[n-1] \cdot in[-n+1] + \dots + \alpha[1] \cdot in[1] + \alpha[0] \cdot in[0]$

【0040】中央処理装置101は係数列 $\alpha$ の先頭アドレスXをアドレス信号として送出する。記憶装置103はアドレス信号を受け取り、対応するデータ $\alpha[n]$ をデータ信号の下位側を用いて中央処理装置101に送出する。オフセット調整装置102はアドレス信号を受け取り、予めオフセットレジスタ201に保持されたオフセット値( $X-Y$ )とアドレス信号を加算器202を用いて加算し、結果Y(即ち、 $X-Y+X$ )をオフセット

付きアドレス信号として記憶装置104に送出する。記憶装置104はオフセット付きアドレス信号に対応するデータ $in[-n]$ をデータ信号の上位側を用いて中央処理装置101に送出する。

【0041】この結果、中央処理装置101は $\alpha[n]$ と $in[-n]$ を同時に得ることができ、数2によれば、右辺の第1項の $\alpha[n] \cdot in[-n]$ の演算に必要なデータを中央処理装置101が確保したことになる。中央処理装置101が2つのデータを用いて積和演算する実施例については後述する。

【0042】次に、中央処理装置101は $\alpha[n-1]$ に対応するアドレス $X+4$ をアドレス信号として送出する。アドレス $X+4$ は中央処理装置101が直前に使用したアドレスXを+4だけインクリメントすることで容易に求められる。

【0043】記憶装置103はアドレス信号を受け取り、対応するデータ $\alpha[n-1]$ をデータ信号の下位側を用いて中央処理装置101に送出する。オフセット調整装置102はアドレス信号を受け取り、予めオフセットレジスタ201に保持されたオフセット値( $X-Y$ )とアドレス信号を加算器202を用いて加算し、結果 $Y+4$ (即ち、 $X-Y+X+4$ )をオフセット付きアドレス信号として記憶装置104に送出する。記憶装置104はオフセット付きアドレス信号に対応するデータ $in[-n+1]$ をデータ信号の上位側を用いて中央処理装置101に送出する。この結果、中央処理装置101は $\alpha[n-1]$ と $in[-n+1]$ を同時に得ることができ、数2によれば、右辺の第2項の $\alpha[n-1] \cdot in[-n+1]$ の演算に必要なデータを中央処理装置101が確保したことになる。同様な手順で数2の右辺の全ての項を中央処理装置101が係数列 $\alpha$ のアドレスをインクリメントすることにより、各項の対になった2つデータを同時に得ることが可能である。この処理過程において、オフセット調整装置102のオフセットレジスタ206に保持したオフセット値( $X-Y$ )は同一の値を使用する。

【0044】最終的に得られた結果 $out[0]$ をデータ列 $out$ の先頭アドレスZに格納する。格納手段として、中央処理装置101はデータ列 $out$ の先頭アドレスZをアドレス信号として送出する。演算結果 $out$ のデータ列は記憶装置104に確保された記憶領域に格納されることになり、この時はライト動作であるから、オフセット調整装置102のセクタ203がアドレスバス110を選択してアドレスZをアドレスバス111に伝達する。アドレスZは第2の記憶装置104に割当てられたアドレスであるから、当該記憶装置104が動作可能にされ、演算結果 $out[0]$ がアドレスZにストアされる。このように、ライト動作では、オフセット調整装置102によるアドレスオフセット加算機能が実質的に無効にされ、CPU101のアクセスアドレスで指定

された記憶装置104の記憶エリアに直接データをストアすることができる。以上の動作手順により、時刻T=0の時のFIRフィルタの畳み込み演算が完了する。

【0045】次に時刻T=1の時の出力データout[1]は数3の式で表される。

【0046】

【数3】  $out[1] = \alpha[n] \cdot in[-n+1] + \alpha[n-1] \cdot in[-n+2] + \dots + \alpha[1] \cdot in[2] + \alpha[0] \cdot in[1]$

【0047】この時のオフセット値は $(x-Y+4)$ となり、out[1]の格納アドレスはZ+4となる。演算手順については、時刻T=0の時と同様であり、数3の右辺の第1項から順次演算する。

【0048】一般的に、係数列 $\alpha$ の先頭アドレスをX、入力データ列inの先頭アドレスをY、出力データ列outの先頭アドレスをZとすると、時刻T=iの時のオフセット値は $X+Y+4i$ となり演算結果out[i]の格納アドレスはZ+4iとなる。このように、時刻Tが刻々変化するとき、時刻の変化に伴う演算に必要なアドレスの変更は上記オフセット値および格納アドレスのみとなり、オーバーヘッドなく処理を行うことが可能である。

【0049】以上のように、FIRフィルタの畳み込み演算に本発明を適用することにより、積和演算に必要な乗数(係数 $\alpha$ )、被乗数(入力データin)を同時に中央処理装置101に取り込むことができ、メモリアクセス処理の回数を半減し、パイプライン処理の乱れを低減することが可能である。

【0050】次に図3を用いてDSPユニット121の積和演算装置の一例について説明する。ここでは前述の図1及び図2を用いて説明したように、1つのアドレス信号に対して2つの異なるデータが同時(同時刻)にデータバス112の上位側、下位側にそれぞれ現れるものとする。

【0051】図3において、301で示されるものは乗算結果を累積加算する加算器、302は乗算器、303はアドレスのインクリメント処理またはデクリメント処理を行うアドレス加算器、304は乗数を格納するレジスタ、305は被乗数を格納するレジスタ、306は乗算結果を格納するレジスタ、307は加算器301の結果を格納するレジスタ、308は直前のアドレスを格納するレジスタである。

【0052】各レジスタの最上位ビットを符号ビットSとし、データは2の補数表示とする。また、データ信号、レジスタ306、レジスタ307およびレジスタ308のビット長を32ビットとし、レジスタ304およびレジスタ305のビット長を16ビットとする。本発明によれば、2つの異なるデータが複数本のデータバス112を二分した上位側、下位側にそれぞれ現れる。そこで、レジスタ304をデータ信号の下位側16ビット

と接続し、レジスタ305をデータ信号の上位側16ビットと接続する。これにより2つの異なるデータを同時に格納し、後段の積和演算処理に必要な乗数、被乗数を得たことになる。

【0053】次に図3の回路の動作を前述のFIRフィルタの畳み込み演算に適用し説明する。係数列 $\alpha$ の先頭アドレスをX、入力データ列inの先頭アドレスをY、出力データ列outの先頭アドレスをZとし、配置場所、配置順序は前述の図1、図2を用いて説明した例に従うものとする。便宜上、図3のレジスタ304をX0、レジスタ305をY0、レジスタ306をM0、レジスタ307をA0、レジスタ308をR0と記述する。時刻T=iにおける積和演算処理の動作を下記に示す。

【0054】〔1〕オフセット調整装置102のオフセットレジスタ201にオフセット値 $(X-Y+4i)$ を設定する。

〔2〕A0を初期化する。

〔3〕R0の値を設定し、 $R0=X$ とする。

〔4〕R0の値をアドレス信号に出力し、各記憶装置より所望のデータを得る。この時、 $\alpha[n]$ がデータ信号の下位側にin[-n+i]がデータ信号の上位側に出力される。

〔5〕X0にデータ信号の下位側データ、即ち $\alpha[n]$ を格納し、Y0にデータ信号の上位側データ、即ちin[-n+i]を格納する。

〔6〕X0とY0を乗算しM0に格納する。

〔7〕A0とM0を加算しA0に格納する。

〔8〕R0の値をインクリメントする。上記〔4〕から〔7〕までの処理をn+1回繰り返す。

〔9〕R0の値を設定し、 $R0=Z$ とする。

〔10〕R0の値をアドレス信号に出力し、A0の値を記憶装置に格納する。このA0の値が時刻T=iの時の出力結果out[i]に相当する。

【0055】以上の動作により、時刻T=iの時のフィルタ処理が完了する。ここで、上記〔4〕から〔5〕までの動作と上記〔6〕から〔8〕までの動作は並列に実行することができる。

【0056】従来、乗数、被乗数の2つのデータの読み出しにはアドレスのインクリメント処理やデクリメント処理用の加算器やレジスタがそれぞれのデータに対して必要になる。しかしながら、本発明の一実施例を示す積和演算回路図では予めオフセット値を設定することにより、上記2つのデータの片方のアドレスを操作するだけで、同時に乗数と被乗数の2つのデータを乗算器に読み込むことが可能である。これにより、連続した積和演算処理を高速に実現することができる。

【0057】図8は図1のマイクロプロセッサを用いて実現したパイプライン処理の一例を示したものである。

命令1から命令7までの各命令は積和演算命令とし、命



令の実行処理(E)は単一サンプリング時間内で完了するものとする。また、命令のフェッチ処理(F)は従来の技術である命令長の縮小化を適応し、2回に1回の割合で発生するものとする。このとき、本発明を適応することにより、各命令毎にメモリアクセス処理(M)回数が1回で済むことから、図7に示す従来のパイプライン処理に比べて60%の高速化に成功している。

【0058】以上本発明者によってなされた発明を実施形態に基づいて具体的に説明したが、本発明はそれに限定されるものではなく、その要旨を逸脱しない範囲において種々変更可能であることは言うまでもない。

【0059】例えば、以上の説明では、本発明をFIRフィルタの畳み込み演算に適用した場合について説明したが、本発明は、相関処理や並び換え処理など少なくとも2つ以上のデータ列を必要とする他の演算にも適用できる。また、記憶装置は2個に限定されない。3個以上のデータを並列演算する必要がある場合には3個以上の記憶装置を用いる。例えば3個の記憶装置を用いる場合、CPUのアクセスアドレスに対して、第1のオフセットアドレスを加算したアドレスと、第2のオフセットアドレスを加算したアドレスとを用意し、CPUのアクセスアドレス、第1の加算アドレス、及び第2の加算アドレスを用いて3個の記憶装置から3種類のデータを並列的にリードアクセスすることができる。

【0060】また、本発明は中央処理装置などのシステムロジックと大容量記憶装置とを1つのチップに集積した大規模LSIにも適用することができる。その場合に、チップ内のバス幅も増加傾向にあり、ここに本発明を適用しても、データ転送効率並びにデータ処理効率の向上に資することができる。

【0061】また、マイクロプロセッサが保有する内部回路モジュールは図1に限定されず、所要の機能に応じて、キャッシュメモリ、ダイレクト・メモリ・アクセス・コントローラ等を適宜追加してもよい。

【0062】また、メモリーネーブル信号はCPUが直接出力してもよい。また、本発明に係るデータ処理装置はマルチチップで構成してもよい。例えば図10に例示されるように、CPUを含むマイクロプロセッサ140及び記憶装置103、104を夫々別々のチップとして用意し、それらをプリントサーキット基板のような回路基板に実装し、オフセット調整装置102をTTL回路で実装基板状に形成する。図10においてオフセット調整装置102はマイクロプロセッサ140に内蔵しない。内蔵すると、マイクロプロセッサのアドレス出力端子が倍増するからである。

【0063】

【発明の効果】本願において開示される発明のうち代表的なものによって得られる効果を簡単に説明すれば下記の通りである。

【0064】すなわち、格納アドレスエリアが逐次変化

される複数の被乗数データと格納アドレスエリアが一定の複数の乗数データとを順番に同一アクセスサイクルでメモリから一緒に読み出すことができ、しかも、乗数データと被乗数データとを個別的に書換えることができる。

【0065】更に、積和演算処理のための乗数データと被乗数データとの読み出しによるパイプライン処理の乱れを低減できると共に、サンプリング時刻毎の被乗数データの書き込み動作を乗数データの配置アドレスとは無関係に行なうことができ、高速な積和演算を実現することができる。

【図面の簡単な説明】

【図1】本発明に係るデータ処理装置の一例であるマイクロプロセッサのブロック図である。

【図2】オフセット調整装置の一例を示すブロック図である。

【図3】DSPユニットの積和演算装置の一例を示すブロック図である。

【図4】乗数及び被乗数を格納した第1及び第2の記憶装置のアドレスマップの一例を示す説明図である。

【図5】一つの記憶装置の一例を示すブロック図である。

【図6】パイプライン処理の一例を示したタイミング図である。

【図7】2つのデータのメモリアクセス処理が必要な場合のパイプライン処理の様子を示すタイミング図である。

【図8】図1のマイクロプロセッサを用いて実現したパイプライン処理の一例を示すタイミング図である。

【図9】FIRフィルタを機能的に説明するためのブロック図である。

【図10】マルチチップで構成したデータ処理システムの一例を示すブロック図である。

【符号の説明】

100 マイクロプロセッサ

101 中央処理装置

102 オフセット調整装置

103 第1の記憶装置

104 第2の記憶装置

121 DSPユニット

201 オフセットレジスタ

202 加算器

203 セレクタ

204 オフセットコントローラ

301 加算器

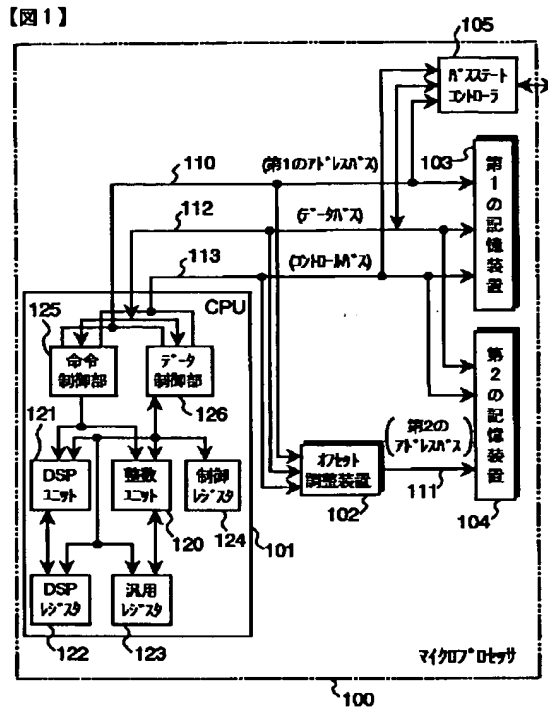
302 乗算器

303 アドレス加算器

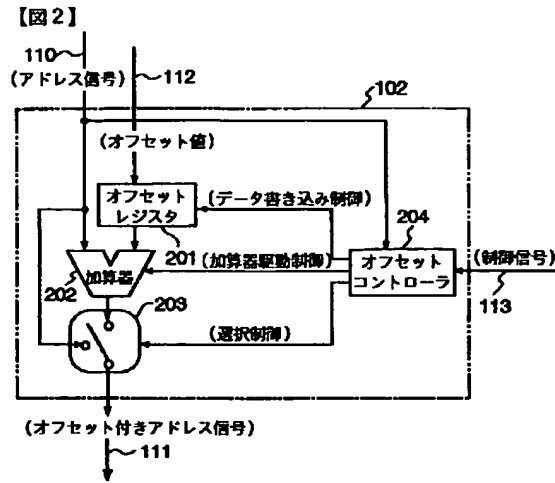
304 乗数格納レジスタ

305 被乗数格納レジスタ

【図1】

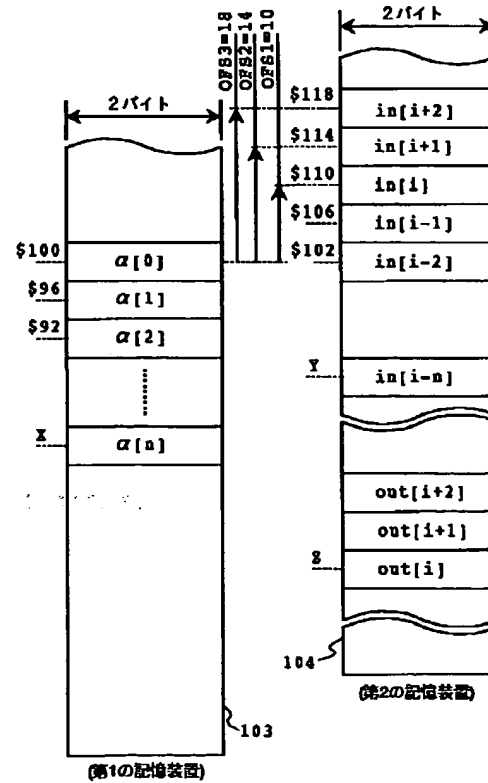


【図2】

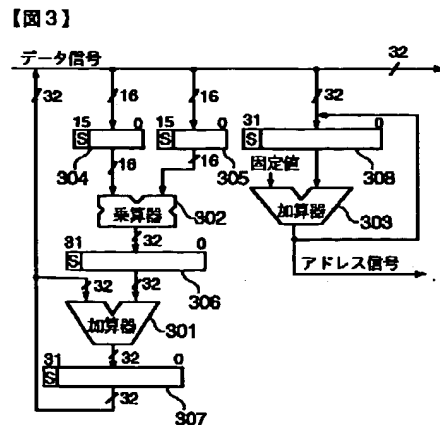


【図4】

【図4】

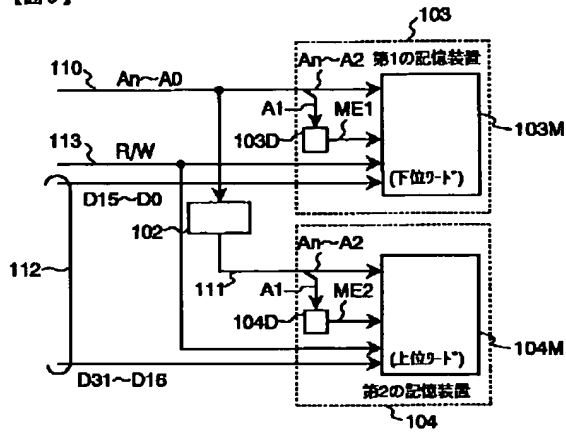


【図3】



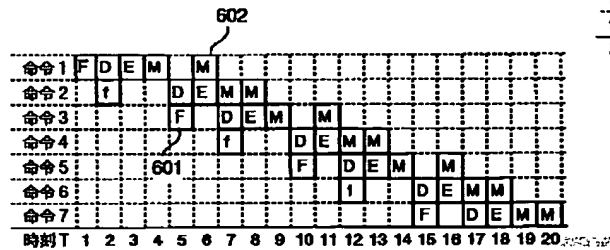
【図5】

【図5】



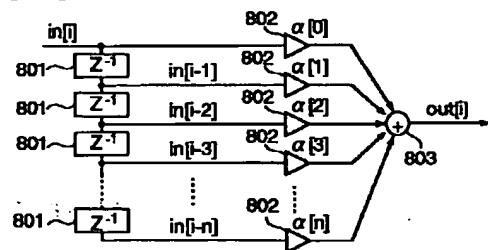
【図7】

【図7】



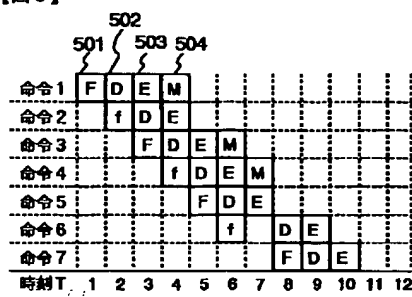
【図9】

【図9】



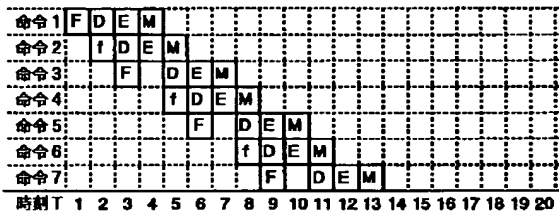
【図6】

【図6】



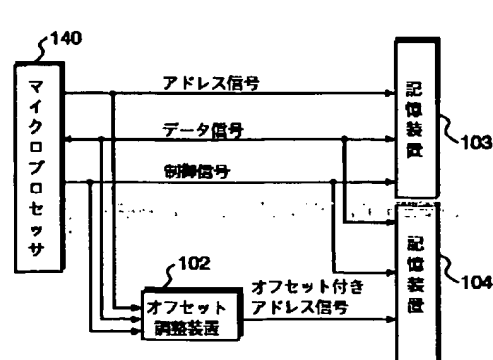
【図8】

【図8】



【図10】

【図10】



フロントページの続き

(72)発明者 石川 敦義

東京都小平市上水本町五丁目20番1号 株  
式会社日立製作所半導体事業本部内

(72)発明者 堤坂 康博

東京都小平市上水本町五丁目20番1号 株  
式会社日立製作所半導体事業本部内

(72)発明者 井上 光義

東京都小平市上水本町5丁目22番1号 株  
式会社日立超エル・エス・アイ・システム  
ズ内

(72)発明者 片桐 雅視

山形県米沢市大字花沢字八木橋東3の3274  
日立米沢電子株式会社内

Fターム(参考) 5B060 CA12 CA14

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**